

Bitwise operations

Bitwise operators perform operations directly on the binary representation of numbers.

ex of bitwise AND(&), OR(|), and NOT(~) operators.

how these bitwise operators work:

- Bitwise AND(&): compares each bit of two numbers. The result is 1 only if both corresponding bits are 1.
- Bitwise OR(|): compares each bit of two numbers. The result is 1 if at least one of the corresponding bits is 1.
- Bitwise NOT(~): A unary operator that inverts all the bits of a single number changing 1 to 0 and 0 to 1.

→ 4 & 2.

The number 4 in binary is 100.

The number 2 in binary is 010.

$$100 \& 010 \Rightarrow 000$$

bitwise 4 & 2. = 0

a) ~4

$$\sim 4 = -(n+1)$$

$$= -5$$

b) 4 / 2

4 in binary 100

2 in binary 010

$$4 \rightarrow 100$$

$$2 \rightarrow 010$$

$$100 / 010 \Rightarrow 110_2$$

$$(1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 4 + 2 + 0 = 6$$

the b7f42

c) 4 / 2

The operation 4 / 2 is a standard division

The result of division is 2.

④ Relational operators

Relational operators compare two values and return a boolean result : 1 for true and 0 for false.

operator	Example	result
$>$ (greater than)	$4 > 2$	1 (true)
$<$ (less than)	$4 < 2$	0 (false)
\geq	$4 \geq 2$	1
\leq	$4 \leq 2$	0
$=$	$4 = 2$	0 (false)
\neq (not equal)	$4 \neq 2$	1

C program for relational operators

```
#include <stdio.h>
int main() {
    int a=4;
    int b=2;
    printf("4 > 2 is : %d\n", a>b);
    printf("4 < 2 is : %d\n", a<b);
    printf("4 >= 2 is : %d\n", a>=b);
    printf("4 <= 2 is : %d\n", a<=b);
    printf("4 == 2 is : %d\n", a==b);
    printf("4 != 2 is : %d\n", a!=b);
    return 0;
}
```